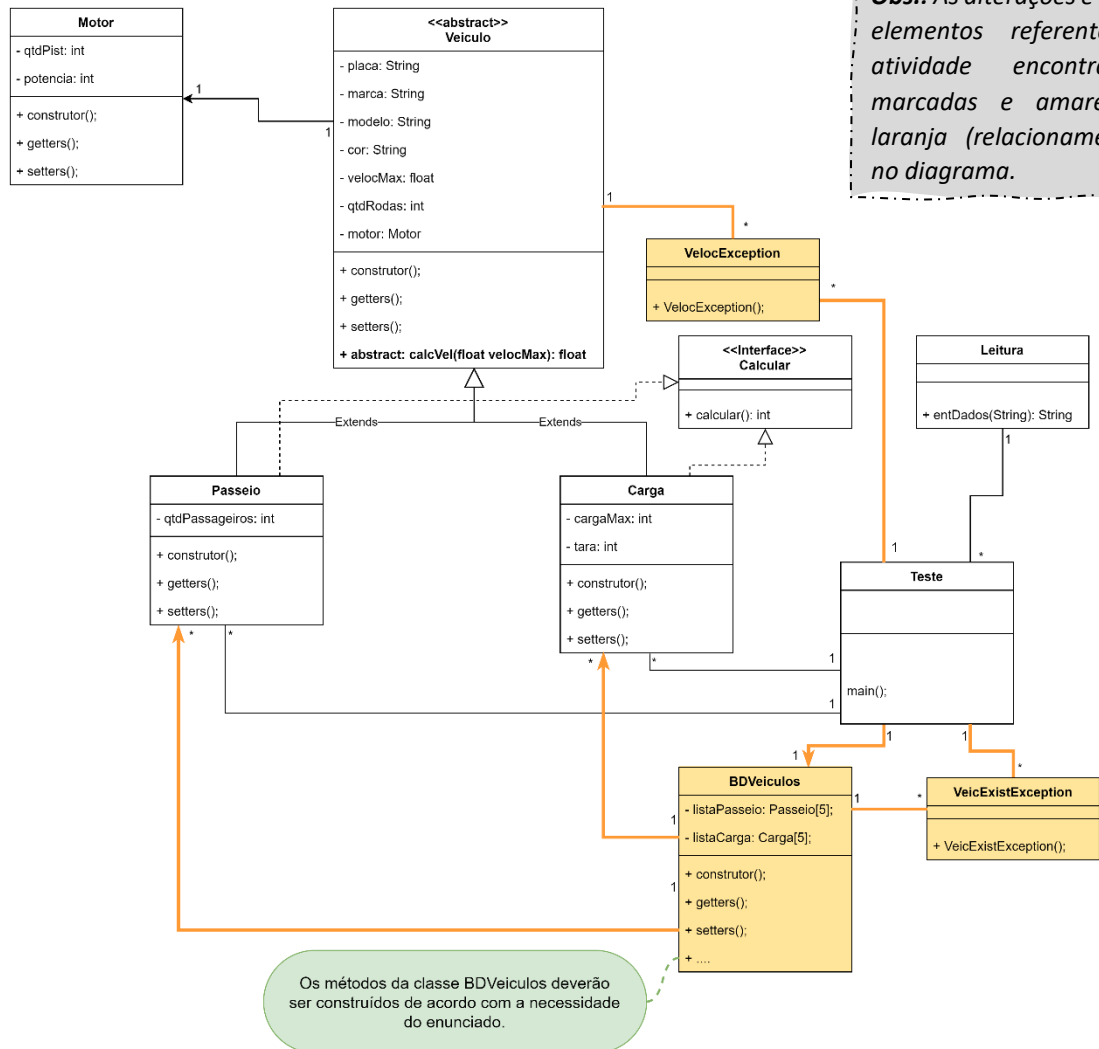


Atividade 06 – Exceções

➔ Este exercício trata-se de uma evolução da Atividade 04.

1) OBSERVE O SEGUINTE DIAGRAMA DE CLASSES



2) UTILIZE O CÓDIGO DESENVOLVIDO NA ATIVIDADE 04 E DESENVOLVA OS NOVOS ELEMENTOS APRESENTADOS EM VERMELHO NO DIAGRAMA ACIMA. ABAIXO, SEGUIE A LISTA DE REQUISITOS A SEREM SEGUIDOS:

NÃO DEVERÁ SER UTILIZADO O RECURSO DE GUI (INTERFACE GRÁFICA)

A **classe Teste** deverá conter o **Menu Inicial** com as seguintes opções (e suas funcionalidade desenvolvidas):

Sistema de Gestão de Veículos - Menu Inicial

1. Cadastrar Veículo de Passeio
2. Cadastrar Veículo de Carga
3. Imprimir Todos os Veículos de Passeio
4. Imprimir Todos os Veículos de Carga
5. Imprimir Veículo de Passeio pela Placa
6. Imprimir Veículo de Carga pela Placa
7. Sair do Sistema

Observações sobre as opções do Menu Inicial:

- **Opções 5 e 6:** O usuário informa a placa do veículo e o programa deve executar uma busca no vetor do respectivo tipo de veículo e imprimir os dados deste; **Caso não exista um veículo com esta placa o sistema deve avisar o usuário e retornar ao Menu Inicial.**
- **Opção 7:** o sistema será encerrado apenas quando o usuário escolher esta opção.

3) O SISTEMA DEVERÁ PERMITIR QUE:

- a. A cada operação realizada no sistema, este retornará ao Menu Inicial;
- b. Armazenar 5 de veículos de cada tipo. Crie 2 vetores, um para os veículos de passeio e outro para os de carga;
- c. Ao inserir um novo veículo o sistema deverá verificar se já existe um veículo cadastrado com a mesma placa. Caso exista, deverá:
 - i. Disparar a exceção **VeicExistException** (que será do tipo verificada);
 - ii. Seu método construtor deverá informar ao usuário sobre esta existência ("Já existe um veículo com esta placa");
 - iii. Voltar ao Menu Inicial.
- d. Sobre a Velocidade Máxima dos veículos:
 - i. Caso a velocidade máxima de um veículo seja menor que 80 ou maior 110, deverá disparar a exceção **"VelocException"** (que será do tipo verificada). Seu método construtor deverá informar que: "A velocidade máxima está fora dos limites brasileiros".
 - ii. Neste caso (após disparar a exceção), se estiver "tentando" atribuir velocidade para um carro de passeio, este deverá assumir 100 (Km/h para velocidade máxima) se for um veículo de carga assumirá 90 Km/h;
- e. Ao final do cadastro de cada veículo, deverá ser perguntado se deseja cadastrar mais um (do mesmo tipo), caso a resposta seja **NÃO** deverá voltar ao Menu Inicial. **Caso a resposta seja SIM, permitir uma nova inserção;**

- f. A velocidade de cada veículo sempre será armazenada em Km/h, porém o método abstrato `calcVel` (sobrescrito nas classes `Passeio` e `Carga`) deverá calcular (não alterando o valor do atributo `velocMax`) e retornar o resultado da seguinte forma:

- i. Se o veículo for de passeio, deverá converter de Km/h para M/h (metros por hora) e retornar este valor;

1 kilometer/hour = 1000 meter/hour

- ii. Caso seja de carga, converterá de Km/h para Cm/h (centímetros por hora) e retornar este valor;

1 kilometer/hour = 100000 centimeter/hour

Use o método `calcVel(float velocMax)`, da classe-mãe, para fazer este cálculo.

Atenção:

- O método `calcVel(float velocMax)` **NÃO** deve alterar o valor do atributo `velocMax`, **apenas convertê-lo** e retornar o valor convertido para que seja exibido na tela por meio da classe `Teste`;

- g. Sempre que se optar pela opção 5 ou pela 6 do Menu Inicial, estes valores serão apresentados junto com as demais informações do respectivo veículo;
- h. O método “calcular” da interface “Calc”, quando sobrescrito:
- i. Na classe `Passeio`: retornará a soma das quantidades de letras existentes em todos os atributos do tipo `String`;
- ii. Na classe `Carga`: retornará a soma de todos os valores contidos nos atributos numéricos.
- i. Perceba que, pelo diagrama não há uma associação entre as classes `Veículo` e `Teste` nem entre `Motor` e `Teste`. Não poderá ser declarado um objeto do tipo `Motor` nas classes: `Passeio`, `Carga` e `Teste`. Somente haverá declaração na classe-mãe `Veiculo`;
- j. Todas entradas e saídas de dados deverão ser feitas na classe `Teste`:
- i. Entrada de dados: **utilize o método `entDados` da classe `Leitura`**;
- ii. Saídas: como achar melhor, porém deve imprimir os dados na mesma sequência dos atributos da classe-mãe `Veiculo` e depois destes os resultados dos métodos “calcular” e “calcVel”;

4) RESTRIÇÕES

- a. **NÃO PERMITA QUE:**
- i. Que seja possível instanciar objetos diretos da classe mãe (`Veiculo`)
- ii. Sobrescrevam os métodos setters da classe `Veiculo`;

- iii. Sobrescrevam os métodos das classes filhas (Passeio e Carga);
- iv. Que as classes filhas (Passeio e Carga) sejam herdadas;

5) O QUE SERÁ AVALIADO

- a. Construção das **novas** classes, com os atributos e métodos conforme descritos no diagrama de classe do item 01.
- b. Relacionamento de herança entre as classes.
- c. Validação do menu, suas opções e restrições.
- d. Lançamento das exceções de acordo com as especificações listadas nos itens 3.c e 3.d.
- e. Demais validação dos itens solicitados nos itens 2, 3 e 4.

Importante!

- Atenha-se aos nomes dos elementos (classes, atributos e métodos) conforme apresentados no diagrama.
- Novos métodos poderão ser criados, caso julgue necessário.
- Os itens avaliados são os solicitados no enunciado. Elementos extras NÃO renderão pontos a mais.
- O não cumprimento do que foi solicitado acarretará no decréscimo da nota de acordo com a gravidade da falta.
- A justificativa para qualquer desconto será colocada, pelo avaliador, no campo de feedback de cada Atividade.